



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/726,067	12/01/2003	Matthew Balint	50235-0840	2564

29989 7590 04/20/2007  
HICKMAN PALERMO TRUONG & BECKER, LLP  
2055 GATEWAY PLACE  
SUITE 550  
SAN JOSE, CA 95110

EXAMINER
----------

WANG, BEN C

ART UNIT	PAPER NUMBER
----------	--------------

2192

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/20/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

# Office Action Summary

Application No.

10/726,067

Applicant(s)

BALINT ET AL.

Examiner

Ben C. Wang

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 01 December 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-78 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-78 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_.

### **DETAILED ACTION**

1. Claims 1-78 are pending in this application and presented for examination.

#### ***Specification Objections***

2. The specification is objected to because the following informalities:
  - "Sun Microsystems", cited in [0007], Line 1, is a registered trademark
  - "The node 305 passes the software version information", cited in [0080], Line 3, should be corrected as "The node 302 passes the software version information"

Appropriate correction is required.

#### ***Claim Objections***

3. Claims 2, 28, and 54 are objected to because the following informalities:
  - "A method as recited in Claim 1, wherein each module has a binary signature<sub>i</sub>.", claim 2, line 1, should be corrected "A method as recited in Claim 1, wherein each module has a binary signature."
  - "A computer-readable medium as recited in Claim 1, wherein each module has a binary signature<sub>i</sub>.", claim 28, line 1, should be corrected "A computer-readable medium as recited in Claim 1, wherein each module has a binary signature."
  - "A apparatus as recited in Claim 53, wherein each module has a binary signature<sub>i</sub>.", claim 54, line 1, should be corrected "A apparatus as recited in Claim 53, wherein each module has a binary signature."

Art Unit: 2192

- “computer readable medium”, cited in [0181], Line 1. Examiner suggests to use “computer recordable storage medium” instead

Appropriate correction is required.

***Claim Rejections – 35 USC § 101***

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 27-52 are rejected under 35 U.S.C 101 because the claims are directed to non-statutory subject matter.

In claim 27, a “computer-readable medium” is being cited, line 1, to include transmission media, light waves, a carrier wave etc., cited in [0181], lines 4, 7, and 13 in the specifications; the claim is directed to a computer program product encoding a computer program. However, Applicant defines “computer-readable medium” to include “a computer data signal embodied in a carrier wave”. Signals and carrier waves do not fall within any class of statutory subject matter, and thus the claim is not limited to statutory subject matter. Please see Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility (1300 OG 142), Annex IV, Section (C) for details.

Art Unit: 2192

6. **As to claims 28-52**, they are merely further recited as the computer program product per se, thus, do not cure the deficiency of base claim 27, and also rejected under 35 U.S.C. 101 as set forth above.

***Claim Rejections – 35 USC § 102(e)***

The following is quotation of 35 U.S.C. 102(e) which form the basis for all obviousness rejections set forth in this office action:

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1-4, 7, 9-10, 12, 15-16, 19-30, 33, 35-36, 38, 41-42, 45-56, 59, 61-61, 64, 67-68, and 71-78 are rejected under 35 U.S.C. 102(e) as being anticipated by G. D. Foster (Pat. No. US 6,675,382 B1) (hereinafter 'Foster')

8. **As to claim 1**, Foster discloses a method of dynamic installation and activation of software packages in a node in a distributed network of nodes, the method comprising the computer-implemented steps of: providing a master node (Fig. 1, element 126 – Server; Col. 6, Lines 8-19 – remote server computer might transmit a requested code for an application program through internet, local network and communication interface); providing software package storage means on said master node for storing software packages and software modules (Col. 12, Lines 43-52 – the software files required for installation may be directly downloaded from the remote server onto the local client system; a set of

Art Unit: 2192

database files to track information pertaining to software packages that have been installed or distributed) that the nodes in the network will be using as well as older versions that are kept for regressing a node back to a previous module or software package version (Col. 10, Line 64 through Col. 11, Line 4 – an older release); wherein a software package (Fig. 2 – Package; Col. 6, Lines 37-41) contains at least one module (Col. 7, Lines 1-9 – payload file contains all files that are required for the installation of computer software) and its associated dependency information (Col. 6, Lines 60-64 – a control file that contains control information pertaining to those files and their dependencies); receiving a software update for a node on said master node (Col. 12, Lines 43-45); wherein the software update contains a software package, or a set of software packages (Abstract; Lines 1-5; Col. 3, Lines 46-51; Fig. 2 – Package; Col. 6, Lines 37-41); storing the software update (Col. 12, Lines 43-45) on said software package storage means (Col. 12, Lines 43-45); wherein said master node notifies said node that a software update is being requested (Col. 6, Lines 8-19); and wherein said master node passes said node identities of software package(s) to be updated (i.e., Col. 8, Lines 34-36) and module dependencies (Col. 8, Lines 22-24).

9. **As to claim 27**, a computer-readable medium carrying one or more sequences of instructions for dynamic installation and activation of software packages in a node in a distributed network of nodes, which instructions, when executed by one or more processors, cause the one or more processors to carry

Art Unit: 2192

out the steps of: providing a master node (Fig. 1, element 126 – Server; Col. 6, Lines 8-19 – remote server computer might transmit a requested code for an application program through internet, local network and communication interface); providing software package storage means on said master node for storing software packages and software modules (Col. 12, Lines 43-52 – the software files required for installation may be directly downloaded from the remote server onto the local client system; a set of database files to track information pertaining to software packages that have been installed or distributed) that the nodes in the network will be using as well as older versions that are kept for regressing a node back to a previous module or software package version (Col. 10, Line 64 through Col. 11, Line 4 – an older release); wherein a software package (Fig. 2 – Package; Col. 6, Lines 37-41) contains at least one module (Col. 7, Lines 1-9 – payload file contains all files that are required for the installation of computer software); receiving a software update for a node on said master node (Col. 12, Lines 43-45); wherein the software update contains a software package, or a set of software packages (Abstract, Lines 1-5; Col. 3, Lines 46-51; Fig. 2 – Package; Col. 6, Lines 37-41); storing the software update (Col. 12, Lines 43-45) on said software package storage means (Col. 12, Lines 43-45); wherein said master node notifies said node that a software update is being requested (Col. 6, Lines 8-19); and wherein said master node passes said node identities of software package(s) to be updated (i.e., Col. 8, Lines 34-36) and module dependencies (Col. 8, Lines 22-24).

10. **As to claim 53**, an apparatus of dynamic installation and activation of software packages in a node in a distributed network of nodes, comprising: a master node (Fig. 1, element 126 – Server; Col. 6, Lines 8-19 – remote server computer might transmit a requested code for an application program through internet, local network and communication interface); software package storage means on said master node for storing software packages and software modules (Col. 12, Lines 43-52 – the software files required for installation may be directly downloaded from the remote server onto the local client system; a set of database files to track information pertaining to software packages that have been installed or distributed) that the nodes in the network will be using as well as older versions that are kept for regressing a node back to a previous module or software package version (Col. 10, Line 64 through Col. 11, Line 4 – an older release); wherein a software package contains (Fig. 2 – Package; Col. 6, Lines 37-41) at least one module (Col. 7, Lines 1-9 – payload file contains all files that are required for the installation of computer software); means for receiving a software update for a node on said master node (Col. 12, Lines 43-45); wherein the software update contains a software package, or a set of software packages (Abstract, Lines 1-5; Col. 3, Lines 46-51; Fig. 2 – Package; Col. 6, Lines 37-41); means for storing the software update (Col. 12, Lines 43-45) on said software package storage means (Col. 12, Lines 43-45); wherein said master node notifies said node that a software update is being requested (Col. 6, Lines 8-19); and wherein said master node passes said node identities of



Art Unit: 2192

software package(s) to be updated (i.e., Col. 8, Lines 34-36) and module dependencies (Col. 8, Lines 22-24).

11. **As to claims 2** (incorporating the rejection in claim 1), **28** (incorporating the rejection in claim 27), **and 54** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein each module has a binary signature (Fig. 2, element 230 – digital signature file; Col. 11, Lines 61 through Col. 12, Line 10).

12. **As to claims 3** (incorporating the rejection in claim 1), **29** (incorporating the rejection in claim 27), **and 55** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein each node has a list of desired characteristics stored on said master node which is compared by said master node to each module in the software update to determine which subset of modules should be sent to a node (i.e., Col. 7, Lines 35-38 – OSVERSION and PLATFORM fields; Col. 8, Lines 34-36; Col. 11, Lines 28-48).

13. **As to claims 4** (incorporating the rejection in claim 1), **30** (incorporating the rejection in claim 27), **and 56** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein said node determines running processes on said node that will be affected by the software update using the module dependencies (Fig. 4, step 410 – Check Dependencies;

Art Unit: 2192

Col. 9, Lines 23-25 – prior to installing package, any dependencies are check as specified by the DÉPENDENCIES field in control file).

14. **As to claims 7** (incorporating the rejection in claim 6), **33** (incorporating the rejection in claim 32), **and 59** (incorporating the rejection in claim 58), Foster discloses a method, a computer-readable, and an apparatus wherein if said master node receives an acceptance from said node then said master node sends appropriate software package(s) (Col. 12, Lines 43-45 – the software files required for installation may be directly downloaded from the remote server onto the local client system) for the software update from said software package storage means (Col. 12, Lines 43-45 – the software files required for installation may be directly downloaded from the remote server onto the local client system) to said node (Col. 12, Lines 43-45).

15. **As to claims 9** (incorporating the rejection in claim 8), **15** (incorporating the rejection in claim 14), **35** (incorporating the rejection in claim 34), **41** (incorporating the rejection in claim 40), **61** (incorporating the rejection in claim 60), **and 67** (incorporating the rejection in claim 66), Foster discloses a method, a computer-readable, and an apparatus wherein said node continues with normal operations and notifies said master node that it has completed the software update; and wherein said master node checks the dependencies of the software package(s) for the software update to ensure that any inter-nodal and intra-node

Art Unit: 2192

dependencies are complete (Fig. 4, step – Check Dependencies; Col. 9, Lines 18-33).

16. **As to claims 10** (incorporating the rejection in claim 9), **16** (incorporating the rejection in claim 15), **36** (incorporating the rejection in claim 35), **42** (incorporating the rejection in claim 41), **62** (incorporating the rejection in claim 61), **and 68** (incorporating the rejection in claim 67), Foster discloses a method, a computer-readable, and an apparatus wherein if there are any discrepancies in the inter-nodal and intra-node dependencies (Fig. 4, step 410 – Check Dependencies; Col. 8, Lines 27-29), then said master node notifies a user (Fig. 4, step 425; Col. 9, Lines 23-28; Col. 10, Lines 10-14).

17. **As to claims 12** (incorporating the rejection in claim 7), **38** (incorporating the rejection in claim 33), **and 64** (incorporating the rejection in claim 59), Foster discloses a method, a computer-readable, and an apparatus wherein said node extracts version information (Col. 8, Lines 17-21) and dependency information (Col. 8, Lines 22-29) from the software package(s) and stores the information in its local persistent storage (Fig. 1, element 112 – Mass Storage; Col. 6, Lines 15-19).

18. **As to claims 19** (incorporating the rejection in claim 18), **45** (incorporating the rejection in claim 44), **and 71** (incorporating the rejection in claim 70), Foster discloses a method, a computer-readable, and an apparatus wherein if the user

Art Unit: 2192

decides to continue updating said node, then said master node forces said node to accept the software update (Col. 10, Lines 40-47).

19. **As to claims 20** (incorporating the rejection in claim 1), **46** (incorporating the rejection in claim 27), **and 72** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein a user initiates a software update by installing an image containing the software update onto said master node (Col. 1, Lines 30-34).

20. **As to claims 21** (incorporating the rejection in claim 20), **47** (incorporating the rejection in claim 46), **and 73** (incorporating the rejection in claim 72), Foster discloses a method, a computer-readable, and an apparatus wherein the user indicates what nodes and which software package(s) are to be updated (Fig. 2 – Package; Col. 6, Lines 37-41).

21. **As to claims 22** (incorporating the rejection in claim 1), **48** (incorporating the rejection in claim 27), **and 74** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein the software update contains a list of nodes to be updated (Col. 1, Lines 7-8, 25-30).

22. **As to claims 23** (incorporating the rejection in claim 1), **49** (incorporating the rejection in claim 27), **and 75** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein the

Art Unit: 2192

software update contains a list of software packages destined for each node (Col. 3, Lines 46-51; Fig. 2 – Package; Col. 6, Lines 37-41).

23. **As to claims 24** (incorporating the rejection in claim 1), **50** (incorporating the rejection in claim 27), **and 76** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein the master node has the ability to categorize nodes into classes where all of the nodes in a particular class of nodes have the same software configuration (i.e., Col. 8, Lines 34-38 - OSVERSION) and may have differing processor types (i.e., Col. 8, Lines 34-38 - PLATFORM).

24. **As to claims 25** (incorporating the rejection in claim 1), **51** (incorporating the rejection in claim 27), **and 77** (incorporating the rejection in claim 53), Foster discloses a method, a computer-readable, and an apparatus wherein a software package contains version information (Col. 8, Lines 17-21), dependency information (Col. 8, Lines 22-24), and other metadata information pertaining to software in the package (Col. 8, Lines 12-55).

25. **As to claims 26** (incorporating the rejection in claim 25), **52** (incorporating the rejection in claim 51), **and 78** (incorporating the rejection in claim 77), Foster discloses a method, a computer-readable, and an apparatus wherein the metadata includes a list of application program interface (API) providers and consumers (Col. 8, Lines 30-32 – MAINTAINER field; Col. 10, Lines 56-60).

***Claim Rejections – 35 USC § 103(a)***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

26. Claims 5-6, 8, 14, 18, 31-32, 34, 40, 44, 57-58, 60, 66, and 70 are rejected under 35 U.S.C. 103(a) as being unpatentable over Foster in view of Oreizy et al., (*Architecture-Based Runtime Software Evolution*, 1998, *IEEE*) (hereinafter 'Oreizy')

27. **As to claims 5** (incorporating the rejection in claim 4), **31** (incorporating the rejection in claim 30), **and 57** (incorporating the rejection in claim 56), Foster discloses software packaging system (Abstract) but does not explicitly disclose a method, a computer-readable, and an apparatus wherein said node notifies affected processes that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the notified processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said node; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said node.

Art Unit: 2192

However, in an analogous art of architecture-based runtime software evolution, Oreizy discloses a method, a computer-readable, and an apparatus wherein said node notifies affected processes that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the notified processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said node; and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said node (Sec. 1 - Introduction, 4<sup>th</sup> Para. – connectors mediate and govern interactions among components, and thereby separate computation from communication, minimize component interdependencies,...; Sec. 4.3 – Runtime Component Replacement, 2<sup>nd</sup> Para. – the model rejects upgraded components when they do not satisfy explicit performance and accuracy requirement).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Oreizy into the Foster's system to further provide a method, a computer-readable, and an apparatus wherein said node notifies affected processes that the software update is being requested; wherein each notified process evaluates the effect that the software update will have on its operation; wherein if any of the notified processes determine that the software update will degrade or have a negative impact on said node's normal operation, the process returns a veto to said node;

Art Unit: 2192

and wherein if a process finds that the software update will have no negative effects, the process returns an acceptance of the software update to said node.

The motivation is that it would advantageously enhance the Foster's system by taking, advancing and/or incorporating Oreizy's system which provides the software architectures which explicitly models connectors that mediate and govern interactions among components, and thereby separate computation from communication, minimize component interdependencies, and facilitate system understanding, analysis, and evolution; the benefits of runtime evolution are from a systematic, principled approach to runtime change supported by a reusable infrastructure as once suggested by Oreizy (i.e., Sec. of Introduction, 1<sup>st</sup> Para., 2<sup>nd</sup> Para., 4<sup>th</sup> Para.).

28. **As to claims 6** (incorporating the rejection in claim 5), **32** (incorporating the rejection in claim 31), **and 58** (incorporating the rejection in claim 57), Oreizy discloses a method, a computer-readable, and an apparatus wherein said node waits for all of the notified processes to return the results of their evaluations and once all of the processes have reported to said node, said node notifies said master node if any of the processes have vetoed the software update (Sec. 1 - Introduction, 4<sup>th</sup> Para. – connectors mediate and govern interactions among components, and thereby separate computation from communication, minimize component interdependencies,...; Sec. 4.3 – Runtime Component Replacement, 2<sup>nd</sup> Para. – the model rejects upgraded components when they do not satisfy explicit performance and accuracy requirement).



29. **As to claims 8** (incorporating the rejection in claim 7), **14** (incorporating the rejection in claim 13), **34** (incorporating the rejection in claim 33), **40** (incorporating the rejection in claim 39), **60** (incorporating the rejection in claim 59), **and 66** (incorporating the rejection in claim 65), Oreizy discloses a method, a computer-readable, and an apparatus wherein said node immediately runs software package modules, by loading the modules from the software package(s) and signals processes that are being replaced by the modules and the affected processes that the changeover is going to occur; wherein when all of the signaled processes indicate that they are ready and waiting for the changeover, said node starts new modules and signals the affected processes that the changeover has occurred; wherein each module starts without affecting normal operation of said node; and wherein each affected process restarts, if required, without affecting normal operation of said node (Sec. 1 - Introduction, 4<sup>th</sup> Para. – connectors mediate and govern interactions among components; and thereby separate computation from communication, minimize component interdependencies,...; Sec. 4.3 – Runtime Component Replacement, 2<sup>nd</sup> Para. – the model rejects upgraded components when they do not satisfy explicit performance and accuracy requirement; Sec. 4.3 – Runtime Component Replacement, 1<sup>st</sup> Para. – the state of the executing component must be transferred to the new component, and both components must not be simultaneously active during the change; corrective and adaptive evolution are characteristic of such changes; Sec. 5.2 – Connectors. 4<sup>th</sup> Para.).

Art Unit: 2192

30. **As to claims 18** (incorporating the rejection in claim 6), **44** (incorporating the rejection in claim 32), **and 70** (incorporating the rejection in claim 58), Oreizy discloses a method, a computer-readable, and an apparatus wherein if said master node receives a veto from said node, then said master node does not update said node (Sec. 1 - Introduction, 4<sup>th</sup> Para. – connectors mediate and govern interactions among components, and thereby separate computation from communication, minimize component interdependencies,...; Sec. 4.3 – Runtime Component Replacement, 2<sup>nd</sup> Para. – the model rejects upgraded components when they do not satisfy explicit performance and accuracy requirement) and notifies a user that the software update will adversely affect said node (Sec. 7 – Tools Supporting Architecture-Based Evolution of Software System, sub-sec of Describing Runtime Change, 2<sup>nd</sup> Para., Lines 12-14); and wherein the user must then make a decision whether to update some or all of the nodes, or to abort the update.

31. Claims 11, 13, 17, 37, 39, 43, 63, 65, and 69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Foster in view of Oreizy and in further view of Moshir et al., (Pub. No. US 2004/0003266 A1) (hereinafter 'Moshir')

32. **As to claims 11** (incorporating the rejection in claim 8), **37** (incorporating the rejection in claim 34) **and 63** (incorporating the rejection in claim 60), Foster and Oreizy do not explicitly disclose a method, a computer-readable, and an

Art Unit: 2192

apparatus wherein when said node does not store the software package(s) in its local persistent storage, then said node can later regress back to previous modules stored in the local persistent storage if it restarts or said master node tells it to regress.

However, in an analogous art of non-invasive automatic offsite patch fingerprinting and updating system and methods, Moshir discloses a method, a computer-readable, and an apparatus wherein when said node does not store the software package(s) in its local persistent storage, then said node can later regress back to previous modules stored in the local persistent storage if it restarts or said master node tells it to regress (Abstract, Lines 6-9 – when a failure is detected, the rollout is stopped and the software can be automatically removed from those computers that already were updated; [0019], Lines 4-7; [0030], Lines 6-13 – if the package has been installed on more than one computer, they all can be removed).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Moshir into the Foster-Oreizy's system to further provide a method, a computer-readable, and an apparatus wherein when said node does not store the software package(s) in its local persistent storage, then said node can later regress back to previous modules stored in the local persistent storage if it restarts or said master node tells it to regress in Foster-Oreizy system.

The motivation is that it would enhance the Foster-Oreizy's system by taking, advancing and/or incorporating Moshir's system which facilitates software

Art Unit: 2192

deployment, software installation, software updating.... Across multiple operating systems and devices, across a network as once suggested by Moshir (i.e., [0020]).

33. **As to claims 13** (incorporating the rejection in claim 12), **39** (incorporating the rejection in claim 38), **and 65** (incorporating the rejection in claim 64), Foster discloses digital signature file (Fig. 2, element 230 – Digital-Signature File) but both Foster and Oreizy do not explicitly disclose method, a computer-readable, and an apparatus wherein said node compares binary signatures of the modules in the software package(s) with corresponding modules stored in the local persistent storage to discover which modules have been updated; wherein any binary signatures that match indicate that the module has not changed; and wherein any modules that have different binary signatures replace the corresponding modules stored in the local persistent storage.

However, in an analogous art of non-invasive automatic offsite patch fingerprinting and updating system and methods, Moshir discloses method, a computer-readable, and an apparatus wherein said node compares binary signatures of the modules in the software package(s) with corresponding modules stored in the local persistent storage to discover which modules have been updated; wherein any binary signatures that match indicate that the module has not changed; and wherein any modules that have different binary signatures replace the corresponding modules stored in the local persistent storage (Fig. 9, elements 908 – Existence Test , 910 – Signature Block; [0090] – an existence

Art Unit: 2192

test which can use the signature block information to determine if a specific patch has been loaded on a machine; [0092]-[0093]; [0106]-[0107]).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Moshir into the Foster-Oreizy's system to further provide method, a computer-readable, and an apparatus wherein said node compares binary signatures of the modules in the software package(s) with corresponding modules stored in the local persistent storage to discover which modules have been updated; wherein any binary signatures that match indicate that the module has not changed; and wherein any modules that have different binary signatures replace the corresponding modules stored in the local persistent storage in Foster-Oreizy system.

The motivation is that it would enhance the Foster-Oreizy's system by taking, advancing and/or incorporating Moshir's system which facilitates software deployment, software installation, software updating.... Across multiple operating systems and devices, across a network as once suggested by Moshir (i.e., [0020]).

34. **As to claims 17** (incorporating the rejection in claim 6), **43** (incorporating the rejection in claim 32), **and 69** (incorporating the rejection in claim 58), Foster and Oreizy do not explicitly disclose a method, a computer-readable, and an apparatus wherein if more than one node was being updated, the software update will not occur if any node vetoes the software update.

However, in an analogous art of non-invasive automatic offsite patch fingerprinting and updating system and methods, Moshir discloses a method, a computer-readable, and an apparatus wherein if more than one node was being updated, the software update will not occur if any node vetoes the software update (Abstract, Lines 6-9 – when a failure is detected, the rollout is stopped and the software can be automatically removed from those computers that already were updated; [0019], Lines 4-7; [0030], Lines 6-13 – if the package has been installed on more than one computer, they all can be removed).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Moshir into the Foster-Oreizy's system to further provide a method, a computer-readable, and an apparatus wherein if more than one node was being updated, the software update will not occur if any node vetoes the software update in Foster-Oreizy system.

The motivation is that it would enhance the Foster-Oreizy's system by taking, advancing and/or incorporating Moshir's system which facilitates software deployment, software installation, software updating.... Across multiple operating systems and devices, across a network as once suggested by Moshir (i.e., [0020]).

### ***Conclusion***

35. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.


- Moshir et al., Non-Invasive Automatic Offsite Patch Fingerprinting and Updating System and Method (Pub. No. US 2002/0100036 A1)
- Kyle et al., Software Installation and Configuration with Specific Role for Target Computer and Identity Indicator for Authorization for Performance of Features (Pat. No. US 7,203,937 B1)
- Lupini et al., Method of Building Dynamic Installation Packages Using a Declarative Authoring Tool (Pub. No. US 2005/0055692 A1)
- Arnaiz et al., Method, System, Apparatus and Program Product for Distribution and Instantiation of Software Upgrades (Pat. No. US 7,080,371 B1)

36. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.


If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



TUAN DAM  
SUPERVISORY PATENT EXAMINER

BCW 

April 10, 2007